Rudra Goel

Lab 03 Report

ECE 2031 L10

15 September 2024

library ieee;

use ieee.std\_logic\_1164.all;

use ieee.std\_logic\_unsigned.all;

-- Describes the device from the outside

entity RPS\_VHDL is

-- Defines the signals coming into and out of the device

port(

R1, P1, S1 : in std\_logic;

R2, P2, S2 : in std\_logic;

W1, W2 : out std\_logic;

E1, E2 : out std\_logic

);

end RPS\_VHDL;

-- Define the internal architecture of the device

architecture Internals of RPS\_VHDL is

-- Create a 6-bit vector -> gives us easy access to inputs

signal all\_inputs : std\_logic\_vector(5 downto 0);

begin

-- "&" is CONCATENATION, not logical AND.

all\_inputs <= R1 & P1 & S1 & R2 & P2 & S2;

-- Using a "selected signal assignment", aka "with/select"

with all\_inputs select W1 <=

'1' when "100001",

'1' when "010100",

'1' when "001010",

'0' when others;

-- Using a "conditional signal assignment", aka "when/else"

W2 <=

'1' when all\_inputs = "100010" else

'1' when all\_inputs = "001100" else

'1' when all\_inputs = "010001" else

'0';

-- Using when/else in a different way

E1 <=

'1' when (R1 = '1') and (S1 = '1') else

'1' when (R1 = '1') and (P1 = '1') else

'1' when (S1 = '1') and (P1 = '1') else

'0';

-- Using Boolean expression

E2 <= (R2 and S2) or (R2 and P2) or (S2 and P2);

end Internals;

**Figure 1.** VHDL code to model a Rock-Paper-Scissor game between two players. “R1”, “S1”, “P1”, “R2”, “S2”, “P2” are inputs corresponding to rock, paper, and scissors for players one and two, respectively. Outputs “W1” & “W2” indicate which player wins. Outputs “E1” & “E2” indicate an invalid input combination for players one and two, respectively.

A screenshot of a computer

Description automatically generated

**Figure 2.** Waveform simulating all possible inputs for Rock-Paper-Scissor game between two players. “W1” goes high when player 1 wins & “W2” goes high when player 2 wins. “E1” and “E2” go high when players one or two put more than one of their dedicated inputs high.

A diagram of a machine

Description automatically generated

**Figure 3.** Schematic implementing with propagation delays for each gate. The greatest propagation delay is 47ns from input “Y” to output “F”.

A screen shot of a graph

Description automatically generated

**Figure 4.** Square waveform of signal with period depending on equation: where . Vertical cursors at each falling edge to measure period of .

A screen shot of a graph

Description automatically generated

**Figure 5.** Oscilloscope capture ofhigh time for square wave in Figure 2 meant to determine duty cycle. Vertical cursors measure high time to be s thus indicating a 25% duty cycle.

A graph with lines and numbers

Description automatically generated

**Figure 6.** Capture of falling edge of signal to determine fall time. Vertical cursors at intersections of 90% of high voltage and 10% high voltage to measure 3ns fall time.

**A graph of a graph

Description automatically generated**

**Figure 7.** High-to-Low propagation delay for input signal (brown) to output signal (green) for two in-series inverters. Vertical cursor at signal intersecting 1.3V measuring 13.9ns high-to-low delay.

**A graph with lines and numbers

Description automatically generated**

**Figure 8.** Low-to-High propagation delay for input signal (brown) to output signal (green) for two in-series inverters. Vertical cursor at signal intersecting 1.3V measuring 14.4ns low-to-high delay.